

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря Сікорського»

Факультет прикладної математики
(повна назва інституту/факультету)

Кафедра системного програмування і спеціалізованих комп'ютерних систем
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ В. П. Тарасенко

« ____ » _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ 6.010203 Комп'ютерна інженерія
(код і назва)

на тему: «Система автоматизації обліку пацієнтів»

Виконав: студент _____ IV курсу, групи К В - 5 2
(шифр групи)

_____ Пилипенко Дмитрій Сергійович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник _____ доцент, к.т.н., Боярінова Ю.Є. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Нормоконтроль _____ доцент, к.т.н., Клятченко Я.М. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній
роботі немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря Сікорського»**

Факультет прикладної математики
(повна назва інституту/факультету)

Кафедра системного програмування і спеціалізованих комп'ютерних систем
(повна назва кафедри)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)

«_____» червня 2019 р.

ЗАВДАННЯ

на дипломний проект студента

Пилипенка Дмитрія Сергійовича

1. Тема проекту: « Система автоматизації обліка пацієнтів», керівник проекту Боярінова Юлія Євгенівна, доц. каф. СПіСКС, к.т.н., затверджені наказом по університету від «22» квітня 2019 р. №1330-С

2. Термін подання студентом проекту «14» червня 2019 р.

3. Вихідні дані до проекту: див. технічне завдання.

4. Зміст пояснювальної записки: аналіз існуючих рішень та обґрунтування теми дипломного проекту; вибір засобів реалізації; особливості розробки програмного забезпечення системи.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Взаємодія модулів системи. Схема структурна; Робота вікна авторизації. Блок-схема; Робота вікна вибору режиму роботи програми. Блок-схема; XML схема зв'язків класів. Схема структурна.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, к.т.н.		

7. Дата видачі завдання: 8.02.2019р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних.	10.02.19	
2	Огляд і аналіз проблемної області.	17.02.19	
3	Постановка задачі.	24.02.19	
4	Проведення огляду і якісний аналізу існуючих аналогічних рішень.	03.03.19	
5	Проектування архітектури системи.	10.03.19	
6	Огляд та якісний аналіз існуючих медичних інформаційних систем.	17.03.19	
7	Проведення огляду і аналізу технічного забезпечення проекту.	24.03.19	
8	Програмна реалізація проекту.	31.03.19	
9	Тестування розробленого ПЗ.	07.04.19	
10	Якісний аналіз готового проекту, робота над помилками та зауваженнями.	21.04.19	
11	Попередній розгляд дипломного проекту на кафедрі	28.05.19	

Студент _____

Пилипенко Д.С.

Керівник роботи _____

Боярінова Ю.Є.

* Консультантом не може бути зазначено керівника дипломного проекту.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
1	A4	ІАЛЦ.045440.002 ТЗ	Система автоматизації обліку пацієнтів	2		
			Технічне завдання			
2	A4	ІАЛЦ.045440.003 ТП	Система автоматизації обліку пацієнтів	2		
			Відомості технічного проекту			
3	A4	ІАЛЦ.045490.0004 ПЗ	Пояснювальна записка	52		
4	A1	ІАЛЦ.045490.0005 Д1	Взаємодія модулів системи. Схема структурна	1		
5	A1	ІАЛЦ.045490.0006 Д2	Робота вікна авторизації Блок-схема	1		
6	A1	ІАЛЦ.045490.0007 Д3	Робота вікна вибору Режиму роботи програми Блок-схема	1		
7	A1	ІАЛЦ.045490.0008 Д4	XML схема зв'язків класів. Схема структурна Блок-схема	1		
8	A4	ІАЛЦ.045490.0009 Д4	Довідка про впровадження ПЗ	1		
9		Диск CD-ROM	Текст пояснювальної записки. Графічні матеріали	1		
ІАЛЦ.045490.0001 ОП						
Змін.		№ докум.	Підпис	Дата		
Розробив		Пилипенко Д.С.				
Перевірив		Боярінова Ю.Є.				
Н. контроль		Клятченко Я.М.				
Затвердив		Тарасенко В.П.				
Система автоматизації обліку пацієнтів. Опис проекту				Літ.	Аркуш	Аркушів
						1
				НТУУ "КПІ" ім. Ігоря Сікорського, ФПМ, КВ-52		

АНОТАЦІЯ

Дипломний проект присвячено дослідженню існуючих медичних інформаційних систем, їх переваг та недоліків. Метою дипломного проекту є створення програмного забезпечення, що є аналогом медичної інформаційної системи для індивідуального користування, яка б відповідала вимогам лікаря-спеціаліста.

Створена система, яка компенсує основні недоліки існуючих медичних інформаційних систем. Також наведено стислий огляд засобів, які використовувались для створення програмного забезпечення, описано головні переваги обраних методів для рішення поставленої задачі.

Система призначена для використання лікарем-хірургом для створення обліку інформації про пацієнта. Інформація зберігається у базі даних, з якої за необхідним запитом можна отримати повну інформацію про пацієнта.

Дипломний проект виконано на 52 аркушах, вона містить перелік посилань на використані джерела з 10 найменувань. У роботі наведено 15 рисунків та 2 таблиці.

Ключові слова: МІС, лікар, система, медичний заклад, документообіг, Java, база даних.

SUMMARY

This graduation thesis is devoted to the study of existing medical information systems, their advantages and disadvantages. The purpose of the graduation thesis is to create software that is analogous to the medical information system for individual use, and that which would meet the requirements of a specialist doctor.

A system is created that compensates for the main disadvantages of existing medical information systems. In addition, a brief overview of the tools used to create the software is shown, and the main advantages of the chosen methods for solving the problem are described.

The system is intended for use by a surgeon with the aim to create a record of the information about the patient. The information is stored in the database, from which you can obtain complete information about the patient upon request.

The graduation thesis is executed on 52 pages; it contains a list of references consisting of 10 titles. The paper presents 15 figures and 2 tables.

Key words: HIS, doctor, system, medical institution, workflow, Java, database.

Зміст

1.	Найменування і область застосування.....	3
2.	Підстава для розробки.....	3
3.	Ціль і призначення роботи.....	3
4.	Джерела роботи.....	3
5.	Технічні вимоги.....	4
	5.1 Функціонал програмного забезпечення.....	4
	5.2 Вимоги до програмного забезпечення.....	4
6.	Вимоги до проєктної документації.....	4
7.	Календарний план.....	5

					ІАЛЦ.045490.002 ТЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Система автоматизації обліку пацієнтів Технічне завдання	Літ.	Аркуш	Аркушів
Розробив		Пилипенко Д.С.					1	4
Перев.		Боярінова Ю.Є.						
Н. контр.		Клятченко Я.М				НТУУ "КПІ" ім. Ігоря Сікорського, ФПМ, КВ-52		
Затвер.		Тарасенко В.П.						

1. НАЙМЕНУВАННЯ ТА ГУЛУЗЬ РОЗРОБКИ

Найменування розробки: Система автоматизації обліку пацієнтів
Галузь застосування: ведення електронного документообігу для індивідуального використання.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

3. ЦІЛЬ ТА ПРЕЗНАЧЕННЯ

Створення програмного забезпечення та дослідження існуючих медичних інформаційних систем.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до продукту, що розробляється:

- Забезпечення надійності збереження даних.
- Забезпечення стабільності роботи програми.
- Простота використання користувачем.
- Швидкодія програми.

					ІАЛЦ.045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

5.2 Вимоги до програмного забезпечення:

- Операційна система на базі Windows.

6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проекту повинна бути розроблена наступна документація:

- Пояснювальна записка.
- Креслення:
 - Схема архітектури програмного забезпечення.
 - Блок-схема роботи вікна авторизації програми.
 - Блок-схема роботи вікна вибору режиму програми.
 - XML схема зв'язків класів.

					ІАЛІЦ.045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних.	10.02.19	
2	Огляд і аналіз проблемної області.	17.02.19	
3	Постановка задачі.	24.02.19	
4	Проведення огляду і якісний аналізу існуючих аналогічних рішень.	03.03.19	
5	Проектування архітектури системи.	10.03.19	
6	Огляд та якісний аналіз існуючих медичних інформаційних систем.	17.03.19	
7	Проведення огляду і аналізу технічного забезпечення проекту.	24.03.19	
8	Програмна реалізація проекту.	31.03.19	
9	Тестування розробленого ПЗ.	07.04.19	
10	Якісний аналіз готового проекту, робота над помилками та зауваженнями.	21.04.19	
11	Попередній розгляд дипломного проекту на кафедрі	28.05.19	

[illegible]

ЗМІСТ

стор.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	2
ВСТУП.....	3
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ, ОБҐРУНТУВАННЯ СТВОРЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ ПАЦІЄНТІВ	4
1.1 Загальний опис проблеми.....	4
1.2 Електронний документообіг.....	9
1.3 Аналіз існуючих систем.....	13
1.4 Постановка задачі.....	19
Висновки до розділу.....	20
2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	21
2.1 Мова програмування Java.....	21
2.2 Комплект розробника JDK.....	22
2.3 IntelliJ IDEA Community Edition.....	22
2.4 Інструментарій JavaFX.....	23
2.5 База даних H2.....	26
2.6 Бібліотека Apache POI.....	29
Висновки до розділу	33
3. ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ ПАЦІЄНТІВ.....	34
3.1 Розроблена архітектура програмного забезпечення.....	34
3.2 Розробка інтерфейсу користувача	34
Висновки до розділу.....	43
ВИСНОВКИ.....	44
ПЕРЕЛІК ПОСИЛАНЬ.....	45
ДОДАТКИ.....	46

					ІАЛЦ.045490.004 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Система автоматизації обліку пацієнтів Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив	Пилипенко Д.С.							
Перев.	Боярінова Ю.Є.						1	55
Н. контр.	Клятченко Я.М					НТУУ "КПІ" ім. Ігоря сікорського, ФПМ, КВ-52		
Затвер.	Тарасенко В.П.							

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API – application programming interface або програмний інтерфейс додатку.

.doc – скорочено від document, текстовий формат для програми текстового процесора WORD

Drag-and-drop (перетягування) — форма виконання певних дій у графічних інтерфейсах користувача, що передбачає використання комп'ютерної миші.

GPU – graphics processing unit або графічний процесор

IDE - Integrated Development Environment або інтегроване середовище розробки

Heap або купа – використовується JRE для виділення пам'яті під об'єкти і класи.

JRE – Java Runtime Environment або середовище виконання для Java, мінімальна реалізація віртуальної машини, необхідних для виконання java-додатків.

Swing – бібліотека для розробки графічних інтерфейсів мовою Java

SWT - Standard Widget Toolkit, бібліотека з відкритим кодом для розробки графічних інтерфейсів мовою Java

МІС – Медична інформаційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

СЕД – система електронного документообігу

СУБД – система управління базами даних

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

ВСТУП

Ефективне інформаційне забезпечення медицини є однією з важливих передумов її якості абсолютно на всіх етапах – від первинного огляду в поліклініці до надання спеціалізованої допомоги в медичних закладах. В нашій країні інформатизація охорони здоров'я означає використання інформаційних технологій, за допомогою яких значно спрощується ведення обліку медичних послуг та управління медичною інформацією, а також надає можливість виконувати необхідні дії в електронному вигляді.

Як показує всесвітня практика – впровадження Медичної інформаційної системи (МІС) та використання її лікарями дозволяє зекономити до 20% часу в порівнянні з використанням лікарями паперових бланків.

В Україні за всю історію її незалежності, було декілька спроб впровадити МІС, починаючи зі спроби в 2006 році, закінчуючи початком введенням МІС в 2017 році. Поки система МІС працює не у всіх медичних закладах країни, лікарі повинні брати складну роботу по заповненню десятків бланків на себе. Крім того впроваджені зараз МІС в основному призначені для використання сімейними лікарями, а не для лікарів - спеціалістів.

Сьогодні існує десятки різних систем, які впроваджуються в тисячах українських медичних закладах, наприклад «EMCIMED», «Helsi», «Dr. Eleks» та інші.

Крім проблеми надійності захисту даних, системи можуть вийти надто «технологічними», адже потребуватимуть обладнання усіх медичних закладів країни комп'ютерами з доступом в інтернет та встановлення на них складного програмного забезпечення.

В даному дипломному проєкті описане програмне забезпечення (ПЗ), що обробляє, зберігає та шифрує данні пацієнтів, які звернулись за допомогою в медичні заклади.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ СИСТЕМА АВТОМАТИЗАЦІЇ ОБЛІКУ ПАЦІЄНТІВ

1.1 Загальний опис проблеми

Медична реформа передбачає використання інформаційних технологій для створення МІС зберігання інформації про пацієнтів. Тому обсяги інформації про пацієнтів, що зберігається в електронному вигляді, зростають. Збільшуються і потреби в функціональному наповненні цих систем.

Незважаючи на те, що зараз масово впроваджуються системи МІС по всій території України, сфера їх застосування наразі охоплює лише первинну медицину – сімейних лікарів, терапевтів і педіатрів. Для решти лікарів проблема паперових носіїв залишається актуальною. Майже половина лікарів досі використовує під час своєї роботи паперові носії.

В Україні, починаючи з 2018 року впроваджується електронна система охорони здоров'я - eHealth, яка дасть змогу забезпечити права пацієнта в частині якості надання медичних послуг, оптимізувати взаємодію пацієнта і лікаря шляхом автоматизації ведення обліку медичних послуг і управлінню медичною інформацією в електронному вигляді.

На даний час система eHealth складається з:

- Центральної бази даних — ЦБД (адміністратор ДП “Електронне здоров’я”);
- Електронних медичних інформаційних систем — МІС (системи, які дають змогу автоматизувати роботу медичних закладів з ЦБД);

Також, центральна база даних eHealth України передбачає забезпечення прозорості фінансування системи охорони здоров'я, надавати можливість працювати без паперу з поступовим переходом на електронний облік, зокрема е-рецепт, е-картка, е-направлення, для створення нових електронних сервісів сформувати бізнес-середовище, створити простір для інновацій в медицині та сприяти розвитку медичного ІТ ринку.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

Швидкий доступ до всієї цієї інформації пацієнта дасть лікарю цілісну картину про його здоров'я, а повна медична історія в електронному вигляді допоможе поставити правильний діагноз. З іншого боку, записатися до лікаря на прийом, перевірити наявність ліків у медичних закладах чи аптеках, отримати рецепт на препарат —це все можливо зробити, не виходячи з дому.

Від початку реформи системи охорони здоров'я за допомогою побудови системи eHealth змінено модель фінансування медичних закладів, ефективному перерозподілу коштів згідно принципу «гроші ходять за пацієнтом» та, відповідно, збільшення доходів медичних працівників.

Ця система дозволить контролювати, наскільки ефективно витрачаються виділені державні кошти. Спочатку, як це зазначалось вище, eHealth охопить первинну медицину - сімейних лікарів, терапевтів і педіатрів. Пацієнти будуть підписувати декларації з вибраними лікарями, а лікарі - реєструвати їх в системі. Це допоможе державі оплачувати лікарю роботу з кожним пацієнтом.

У 2019 році Україна знаходиться на завершальному етапі впровадження першої частини електронної системи охорони здоров'я, зокрема завершена реєстрація медичних установ та майже підписні декларації лікарів і пацієнтів.

Водночас, з боку влади постійно приймаються нормативно-правові акти щодо ухвалення Концепції розвитку екстреної допомоги. Так, в поточному році розпочалась трансформація екстреної медичної допомоги, яка дозволить принципово покращити якість та доступність екстреної медичної допомоги, щоб в критичній ситуації кожен українець оперативно отримував необхідну допомогу.

У зв'язку з впровадженням наступного етапу реформи відбудуться оновлення алгоритмів диспетчеризації і створення нової IT-інфраструктури, що дасть змогу мати інформацію про стан кожного виклику в будь-який час.

Також дуже важливо підтримати на державному рівні престижність професій працівників екстреної медичної допомоги і забезпечити достойну

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

оплату їхньої праці, облаштування сучасних диспетчерських служб по усій країні, шляхом:

- закупівлі спеціалізованого санітарного транспорту;
- оснащення для навчально-тренувальних відділів центрів екстреної медичної допомоги та медицини катастроф (ЦЕМД);
- апаратно-програмного комплексу диспетчерських служб центрів екстреної медичної допомоги та медицини катастроф.

Планується, що до кінця 2019 року в усіх регіонах України діятиме мережа сучасних диспетчерських служб, об'єднаних єдиною електронною системою. Завдяки сучасним диспетчерським, новим протоколам та центральній інформаційно-аналітичній системі, виклики оброблятимуть оперативніше, а бригади екстреної медичної допомоги швидше доїжджатимуть на місце події[1].

Водночас, існують актуальні проблеми, що вповільнюють запровадження системи, зокрема, відсутність єдиного унікального ідентифікатора для пацієнтів, недосконалість інформаційної інфраструктури та взаємодії між загальнодержавними реєстрами, недосконалість ряду реєстрів, недостатність фахових спеціалістів для автоматизації та управління змінами, недостатність комп'ютерного та мережевого обладнання в закладах охорони здоров'я тощо.

Зараз використовуються і впроваджуються лише окремі елементи eHealth. Досі немає розробленої та затвердженої концепції та Національної стратегії електронної системи охорони здоров'я, чіткої затвердженої на законодавчому рівні архітектури системи та технічного завдання. Відсутні гармонізовані міжнародні та затверджені чисельні стандарти збереження, обробки та передачі медичних даних, класифікаторів та довідників. Немає затверджених єдиних функціональних вимог до медичних інформаційних систем (приватних компаній), які б проходили відповідну цим вимогам

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

сертифікацію в залежності від рівня надання медичних послуг та після цього рекомендувалися для впровадження в медичних закладах відповідного рівня.

Сьогодні кожен лікар, окрім картки пацієнта, заповнює цілу низку іншої документації. До того ж, це зовсім не ефективно.

Проте на даний час не скасовано заповнення усіх паперових форм, оскільки серед них є статистичні дані, які потрібні для порівняння.

Також сьогодні не можна перевірити процес заповнення певних форм і те, наскільки дані, записані там, відповідають дійсності. При веденні всієї документації у електронній формі, це можна буде відслідковувати.

Підписуючи декларації з пацієнтами, сімейні лікарі, терапевти і педіатри працюють з персональними даними. Слід зазначити, що важливо, яким буде захист персональних даних пацієнтів.

Усі відомості або сукупність відомостей про пацієнта, які вносяться в декларацію про вибір лікаря, є персональними даними пацієнта (ПІБ, дата народження, реєстраційний номер облікової картки платника податків, номер та серія паспорту або інших документів, що посвідчують особу, адреса проживання та інші дані, за якими можна ідентифікувати пацієнта).

Закон України «Про захист персональних даних» від 1 червня 2010 року № 2297-VI[2] (далі – Закон) регулює правові відносини, пов'язані із захистом і обробкою персональних даних, і спрямований на захист основоположних прав і свобод людини і громадянина, зокрема права на невтручання в особисте життя, у зв'язку з обробкою персональних даних.

Цей Закон поширюється на діяльність з обробки персональних даних, яка здійснюється повністю або частково із застосуванням автоматизованих засобів, а також на обробку персональних даних, що містяться у картотеці чи призначені до внесення до картотеки, із застосуванням неавтоматизованих засобів.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

Згідно з формулюванням Закону персональні дані — відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована.

Отже, коли у системі почнуть працювати електронний лікарняний та електронна медична картка, персональні дані будуть оброблятися для забезпечення лікувального процесу та для покращення функціонування електронної системи охорони здоров'я.

Пацієнт, підписуючи декларацію з лікарем, людина погоджується, що його персональні дані в електронній системі будуть доступні для обробки лікарями, до яких він буде звертатися за медичною допомогою.

На персональні дані пацієнтів має поширюватись дія законодавства про лікарську таємницю і вони повинні забезпечувати захист таких персональних даних. Медичні працівники зобов'язані не допускати розголошення у будь-який спосіб персональних даних, які їм було довірено або які стали відомі у зв'язку з виконанням професійних чи службових або трудових обов'язків, крім випадків, передбачених законом.

Доступ до даних про пацієнта, що містяться в електронній системі охорони здоров'я, можливий лише у разі отримання згоди такого пацієнта (його законного представника) у письмовій формі або у формі, що дає змогу зробити висновок про надання згоди.

Без згоди доступ до інформації про пацієнта можливий лише у таких випадках:

- наявності ознак прямої загрози життю пацієнта;
- за умови неможливості отримання згоди такого пацієнта чи його законних представників;
- за рішенням суду.

За недодержання встановленого законодавством порядку захисту персональних даних, що призвело до незаконного доступу до них або порушення прав пацієнта як суб'єкта персональних даних, передбачена

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

адміністративна відповідальність. А за порушення недоторканності приватного життя (незаконне збирання, зберігання, використання, знищення, поширення конфіденційної інформації про особу або незаконна зміна такої інформації) — кримінальна відповідальність.

Отже, електронна система має бути спроектована для роботи з персональними даними з дотриманням найкращих світових практик у сфері захисту даних, мати комплексну систему захисту інформації (КСЗІ) , а для перевірки надійності захисту даних пройти тестування щодо кібербезпеки та обов’язкову атестацію у Державній службі спеціального зв’язку та захисту інформації (ДССЗІ).

Варто відмітити, що сучасні МІС впроваджуються на рівні медичних закладів, тож їх використання окремими лікарями є майже неможливою. Тому варто створити таку МІС, яка б повністю задовольняла вимоги тих лікарів, для яких наразі ще не створили робочу систему. Тому виникає необхідність створення додаткових засобів, які б допомагали лікарю заповнювати електронні документи обліку пацієнтів з витратою меншого часу.

Проблема, що розглянута в даному дипломному проєкті полягає в тому, що необхідно створити ПЗ для МІС, яка буде допомагати лікарю – хірургу вести облік пацієнтів у електронному вигляді, створювати «історію хвороби» та всі необхідні для медичного закладу документи. Відповідно до цього треба розглянути, які вимоги є до ПЗ, системи електронного документообігу, щоб розроблене ПЗ відповідало всім необхідним вимогам та буде підходити для виконання поставленої задачі.

1.2 Електронний документообіг

З моменту активного розвитку інформаційних технологій в світі, провідну роль відіграють принципи створення, збереження та передавання електронної інформації. Системи електронного документообігу (СЕД) надають змогу значно зекономити час роботи персоналу в порівнянні з

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

паперовими документами та надають можливість автоматизованої обробки даних.

Що стосується сфери охорони здоров'я, можна стверджувати, що одужання пацієнта напряду залежить від своєчасного та коректного діагнозу. Варто відмітити, що при наявності систем електронного документообігу, пацієнт отримує набагато більше уваги від лікаря, оскільки останній більше не витрачає значну частину свого робочого часу на паперову тяганину, а більше приділяє увагу саме лікуванню пацієнта.

Перехід на електронну систему обслуговування сприяє раціоналізації роботи медичних установ, та переходу на наступний рівень надання послуг.

На сьогодні законодавчо передбачені загальні вимоги до інформації та документів у електронній системі охорони здоров'я. Створення, внесення, перегляд інформації та документів у центральній базі даних, внесення змін та доповнень до них здійснюються користувачами відповідно до прав доступу. Інформація та документи створюються та вносяться до центральної бази даних українською мовою. У разі коли використання літер української абетки призводить до спотворення інформації, можуть використовуватися латинські літери і спеціальні символи, зокрема для запису адреси в Інтернеті та адреси електронної пошти.

Документообіг у електронній системі охорони здоров'я здійснюється відповідно до вимог законодавства про електронні документи та електронний документообіг. На всі електронні документи, що вносяться до центральної бази даних, накладається електронний підпис автора або підпис, прирівняний до власноручного підпису відповідно до закону. Інформація під час обробки та обміну в електронній системі охорони здоров'я повинна зберігати цілісність, що забезпечується шляхом захисту від несанкціонованих дій, які можуть призвести до її випадкової або умисної модифікації чи знищення, зокрема шляхом накладення електронного підпису автора або підпису, прирівняного до власноручного підпису відповідно до закону.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

Для внесення інформації та документів до центральної бази даних використовуються державні класифікатори, номенклатури та довідники, затверджені в установленому законодавством порядку, зокрема спеціальні класифікації та переліки. Адміністратор здійснює технічну підтримку застосування таких класифікацій, номенклатури, довідників та переліків. Про кожен документ та інформацію, що внесені до центральної бази даних, автоматично робиться унікальний запис у відповідному реєстрі. Ідентифікатором фізичної особи є унікальний номер запису в Єдиному державному демографічному реєстрі[3]

Варто відмітити, що переваги від використання електронного документообігу досить суттєві. По-перше впровадження таких систем дозволяє економити кошти, призначені для купівлі копіювально-розмножувальної техніки й витратних матеріалів до неї, її ремонт тощо. Також це зменшує витрати на зберігання паперових документів(зокрема, фізичне звільнення місця для документів тимчасового зберігання).

По-друге, ще одним позитивним ефектом від переходу на електронний документообіг є значна економія робочого часу працівників, адже за паперового документообігу вони б витрачали більше часу на ручну обробку даних і виконання обслуговуючих функцій.

Вагомими аргументами на користь електронного документообігу є:

- підвищення швидкості пошуку документів;
- запобігання втрати важливої інформації через недбалість персоналу;
- підвищення рівня інформаційної безпеки, за рахунок шифрування даних;
- централізоване зберігання інформації в електронному архіві.

Однак в системах електронного документообігу існують і свої недоліки. До основних недоліків належать:

- досить високі початкові витрати на впровадження СЕД;

- досить відчутні часові витрати: перехід на електронний документообіг неможливо здійснити за короткий проміжок часу;

- необхідність навчання і перепідготовки персоналу для роботи з СЕД;

- можливість втрати чи витоку інформації

Функціональні можливості систем електронного документообігу представлені в таблиці 1.1

Таблиця 1.1 – Функціональні можливості систем електронного документообігу

Функції	Зміст
Реєстрація документів	Реєстрація і облік
	Ведення журналу реєстрації та обліку
Контроль виконання	Ведення контрольних завдань
	Перенесення термінів виконання завдань при необхідності
	Ведення стану виконання
Пошук	Атрибутивний пошук об'єктів документообігу
Звітність	Атрибутивний пошук об'єктів документообігу
	Журнал реєстрації подій
Адміністрування	Управління процесами
	Налагодження системи
	Розмежування повноважень

Можна зробити висновок, що впровадження електронного документообігу у лікарнях країни є обов'язковим, оскільки дозволить значно зменшити час, який потрібен лікарям для заповнення всіх потрібних бланків, актів, виписок тощо. Варто відмітити, що можлива втрата чи виток інформації є значною проблемою впровадження даної системи.

1.2 Аналіз існуючих систем

eHealth – це сукупність інформаційних сервісів в галузі охорони здоров'я та охоплює інформаційних простір різних галузей охорони здоров'я – медичну практику, управління медичними закладами, медичне право, фармацевтику, інформаційні сервіси для пацієнтів тощо.

Центральний компонент – це спеціальне програмне забезпечення, яке накопичує інформацію, що надходить з периферійного компоненту eHealth – медичних інформаційних систем або іншого програмного забезпечення – за певними правилами.

Периферійний компонент – це програмне забезпечення, зокрема медичні інформаційні системи, з якими безпосередньо працюють медичні працівники в медичних лікувальних закладах на місцях, до яких відносяться «EMCIMED», «Helsi», «Dr. Eleks».

Наразі існує купа МІС які доступні на ринку продуктів всім бажачим, деякі з них являються локальними проєктами вузького спектру застосування, деякі з них являються глобальними проєктами державного значення.

Для аналізу МІС були взяті наступні програмні продукти «EMCIMED», «Helsi», «Dr. Eleks». Для порівняння були взяті саме ці системи, оскільки вони є одними з найрозповсюдженішими МІС в Україні.

Так, за допомогою інформаційної системи «Dr. Eleks» здійснюється реєстрація медичних закладів, лікарів первинної ланки, пацієнтів та фінансування медичного закладу, виходячи із кількості пацієнтів.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

В медичній інформаційній системі «Helsi» надається сервіс запису до лікарів та зберігання медичних даних.

В медичній інформаційній системі «EMCiMED» зосереджені всі необхідні інструменти для роботи медичних працівників з центральним компонентом eHealth, а головне: електронна медична картка пацієнта, персональні дані пацієнта та його медичні записи.

На сьогодні на ринку представлені МІС проходять тестування у сімейних лікарів, терапевтів і педіатрів тощо.

Разом з цим, на жаль є купа лікарів, для яких МІС ще не адаптоване. Лікарі-хірурги - вони досі в очікуванні МІС, котра зможе задовольняти всі їх потреби. Поки на ринку немає таких МІС, лікарі змушені витратити свій дорогоцінний час на роботу з паперовими актами, бланками, історіями хвороби.

Варто відмітити, що більша частина МІС, які представлені на ринку, не мають у відкритому доступі технічних характеристик продукту, а мають лише загальні відомості про продукт. Цей фактор перешкоджає первинному аналізу та порівнянню декількох МІС між собою. Щоб обрати оптимальний варіант потрібно витратити кучу часу на телефонні дзвінки в компанії-постачальники, а також провести не одну особисту зустріч з їх представниками. У відкритому доступі вдалось знайти лише повні технічні характеристики системи «Helsi» [4] та деякі технічні характеристики системи «Dr.Eleks»[5]. Що стосується системи «EMCiMED», тут все набагато складніше. У відкритому доступі майже немає технічних характеристик даного програмного забезпечення, що є, на мою думку, не допустимим для ПЗ такого масштабу. Вся знайдена інформація знаходиться на офіційному сайті системи «EMCiMED» [6].

МІС – досить складні системи, якщо розібратися в графічному інтерфейсі можна за допомогою інструкцій, які надаються до кожної системи, то самостійне встановлення та їх налаштування є неможливим. При

первинному налаштуванні програми в медичних закладах обов'язково потрібен спеціаліст з команди розробників даного програмного продукту.

Деякі МІС пропонують динамічний графічний інтерфейс. Це означає, що за допомогою спеціаліста можна побудувати графічний інтерфейс, який максимально задовольняв вимоги конкретного лікаря. Це є значним плюсом для даних систем, адже від зручності залежить якість і швидкість роботи лікаря. Нажаль МІС в яких присутній динамічний інтерфейс це скоріше виняток з правил. Під час аналізу медичних інформаційних систем, було знайдено лише одну систему, яка має динамічний інтерфейс.

Варто відмітити, що в разі збою в роботі МІС, самостійне рішення проблеми є майже неможливим, оскільки може призвести до ще гірших наслідків. В таких випадках викликається спеціаліст з команди розробників системи, котрий розбирається і вирішує проблему. В свою чергу це є прямою загрозою анонімності пацієнтів.

Для порівняння було взято характеристики програм, оскільки порівняння кількісних характеристик є досить важким, враховуючи їх кількість та складність доступу до технічних характеристик певних МІС

Отже для порівняння було взято наступні характеристики: використання СУБД, надійність, робочий інтерфейс. Всі данні заведено у таблицю 1.2

Таблиця 1.2 –Аналіз існуючих систем

	EMCIMED	Helsi	Dr. Eleks
Використання СУБД	Побудована на базі систем керування базами даних MS SQL	PostgreSQL версії не нижче 9.5 або аналог, MongoDB версії не нижче 3.6 або аналог.	Побудована на основі платформи Microsoft SQL Server 2005/ 2008/ 2012/ 2014.

Продовження таблиці 1.2 – Аналіз існуючих систем

Надійність	Для зберігання і передачі необхідної інформації на робочі місця медперсоналу використовується трирівнева система: SQL-Сервер баз даних - Application Server - Client.	Дані зберігаються у дата-центрі, який отримав сертифікат комплексної системи захисту інформації від Державної служби спеціального зв'язку і захисту інформації України.	Для безпечної комунікації в мережі використовується інфраструктура безпеки доменів Windows із застосуванням служби Active Directory.
Робочий інтерфейс	Див. рис. 1.3	Див. рис. 1.4	Див. рис. 1.5

Добавление госпитализации пациента

Информация о пациенте

№ ЭМК: № ЭМК Ф.И.О. Пациента (Ctrl +1):

Фамилия: Пол: Телефоны:

Имя:

Отчество:

Пол:

Дата рожд.:

Место постоянного проживания

Житель:

Адрес (F5):

Обновить:

Телефоны

Мобильный:

Домашний:

Рабочий:

Другой:

Пример
+7 495 755-55-55
+38 067 123-45-67
045 923-11-30
240-12-34

Место работы:

☐ Страховка ☐ Принадлежность к статусу (инвалид войны, участник войны, пострадавший в результате аварии на ЧАЭС)

☐ Отказ в предоставлении инф-ции

☐ из списка ☒ новый

Дата госпитализации: № карточки:

Отделение поступления:

Тип госпитализации:

В текущем году по поводу данной болезни госпитализирован:

Повышенная чувствительность или непереносимость препарата:

Кем направлен больной:

По уходу за ребенком: ☐

Диагноз лечебного учреждения, направившего больного:

С: 12.09.2013 16:52:05 ЖУРАВЛЕВА А.Н.

Рисунок 1.3 – Интерфейс МІС «EMCIMED»

На прийомі: Мартинюк О. С. 23.11.2018 13:15

1 Прийом (1) 2 Медзаписи 3 **Діагноз** 4 План

3 **ДІАГНОЗ (КЛІНІЧЕСКИЙ)**

Тип діагнозу:

Основне захворювання

Виберіть захворювання:

Код за МКХ-10:

Характеристики захворювання:

Дія з випадком:

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 1.4 – Интерфейс МІС «Helsi»

Зм.	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

17

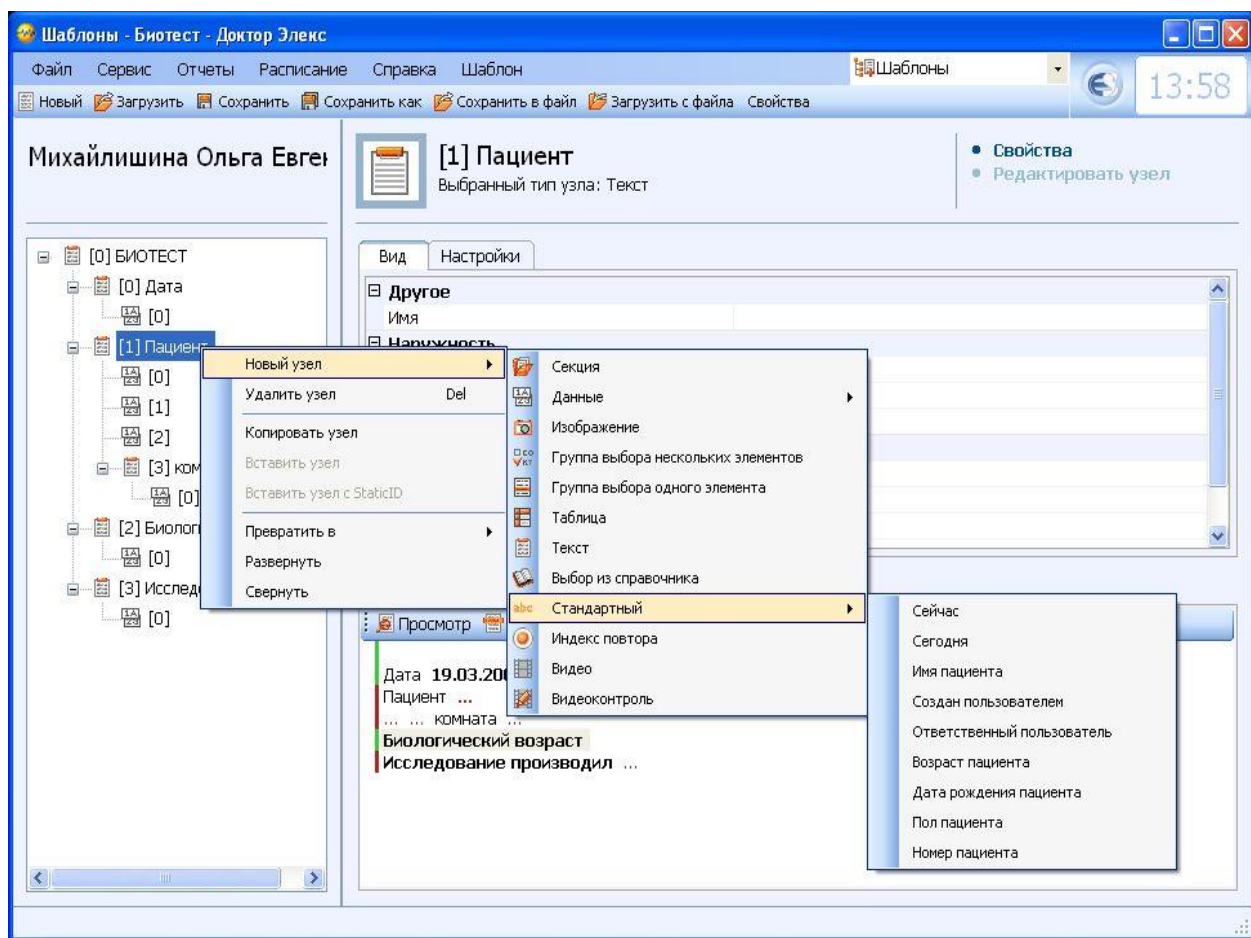


Рисунок 1.5 – Интерфейс МІС «Dr. Eleks»

Проаналізувавши приведені данні можна прийти до висновку, що з існуючих систем, МІС «Helsi» являється оптимальною та має найменшу кількість мінусів з систем, представлених для порівняння. До плюсів можна віднести наявність у вільному доступі технічну характеристику продукту, сертифікат безпеки від Державної служби зв'язку і захисту. Гарним плюсом є те, що МІС «Helsi» існує в мобільній версії. Варто зауважити, що МІС «Helsi» має найкращий та зрозумілий інтерфейс серед усіх представлених систем

Однак, варто відмітити, що кожна з наведених для порівняння систем має свої мінуси. Наразі не розроблена така МІС, яка зможе замінити паперовий документообіг на електронний документообіг для всіх лікарів та лікарень, оскільки багато потрібних функцій ще не реалізовані в цих системах.

Питання збереження інформації досить гостро стоїть, особливо в сфері охорони здоров'я. До тих пір, поки компанії не можуть дати стовідсоткову

гарантію безвідмовної роботи своїх систем, існує висока ймовірність втрати чи витоку інформації.

Таким чином, необхідно створити систему, яка б принаймні компенсувала мінуси вже існуючих систем, була зрозумілою в освоєнні та розроблена з дотриманням усіх державних стандартів та побажань лікарів.

1.4 Постановка задачі

Головна мета дипломного проєкту – створити додаток для існуючих МІС для індивідуального використання, враховуючи потреби та побажання лікарів-хірурга, яка б виступала в якості системи електронного документообігу.

Проаналізувавши всі доступні продукти на ринку, врахувавши всі їх переваги та недоліки було виділено основні програмні вимоги для реалізації цього проєкту.

Ключовими вимогами є:

- Надійність збереження даних
- Зручність в користуванні
- Безвідмовна робота
- Швидкодія програми
- Простий та зрозумілий інтерфейс

Висновки до розділу

Наразі в нашій країні існує багато медичних інформаційних систем, які використовуються в багатьох медичних закладах. Велика кількість лікарів використовують такі системи в своїй роботі, що значно економить їх час та дозволяє приділяти набагато більше уваги пацієнтам. Серед існуючих медичних інформаційних систем багато лікарів зможуть знайти таку, яка буде відповідати їх вимогам.

Серед проаналізованих систем варто звернути увагу на медичну інформаційну систему «Helsi». Згідно наведеного аналізу, ця система являється ідеальним варіантом для лікарів первинної медицини.

Але все ще є велика кількість лікарів, які не мають змоги використовувати МІС під час своєї роботи через наступні причини:

- Відсутність обладнання (ПК, доступ до інтернету)
- Відсутність універсальних МІС для всіх галузів медицини
- Складність налаштування існуючих МІС
- Велика вартість ПЗ

Враховуючи вищесказане, потрібно або допрацьовувати існуючі МІС, або створити нове ПЗ, яке буде виконувати функції МІС в тих галузях медицини, для яких сучасні системи наразі непридатні.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Мова програмування Java

Java — об'єктно-орієнтована мова програмування, яка була розроблена компанією «Sun Microsystems» в 1995 році, і являється основним компонентом платформи Java[7]. В 2009 році компанія «Sun Microsystems» була придбана компанією «Oracle». З 2009 року компанія «Oracle» займається розробкою та підтримкою компонентів для мови програмування Java.

Програми написані мовою Java компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

При розробки мови програмування Java було частково запозичено синтаксис із таких мов програмування як C і C++. Зокрема, взято за основу а потім модифіковано об'єктну модель мови програмування C++. Було усунуто можливість появи деяких конфліктних ситуацій, які могли виникнути в результаті помилки програміста. Також був полегшений сам процес розробки об'єктно-орієнтованих програм за рахунок того, що в мовах програмування C/C++ був ряд дій, котрі повинні були здійснювати програмісти, в мові програмування Java деякі дії доручено виконувати віртуальній машині.

Основна особливість мови програмування Java — платформо-незалежність, тобто незалежність віртуальної машини від апаратного устаткування та системи. Для Java існує декілька десятків різних програмних мов, одні з яких компілюються, а інші працюють безпосередньо із віртуальною машиною. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Програма, що була написана мовою Java, коректно працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому коді та перекомпіляції.

Java використовує автоматичний збирач сміття для керування пам'яттю під час життєвого циклу об'єкта. Програміст вирішує, коли створювати об'єкти, а віртуальна машина відповідальна за звільнення пам'яті після того, як

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		21

об'єкт стає непотрібним. Коли до певного об'єкта вже не залишається посилань, збирач сміття може автоматично прибирати його із пам'яті. Проте, витік пам'яті все ж може статися, якщо код, написаний програмістом, має посилання на вже непотрібні об'єкти, наприклад на об'єкти, що зберігаються у діючих комірках пам'яті.

Збирання сміття дозволене у будь-який час. В ідеалі воно відбувається під час бездіяльності програми. Збірка сміття автоматично виконується при нестачі вільної пам'яті в купі – Heap[8] – для розміщення нового об'єкта, що може призводити до кілька секундного зависання. Тому існують реалізації віртуальної машини Java з прибиральником сміття, спеціально створеним для програмування систем реального часу.

2.2 Комплект розробника JDK

JDK або Java Development Kit[9] – безкоштовний розповсюджуваний компанією «Oracle» комплект розробника застосунків на мові Java, який включає до себе наступні компоненти:

- Компілятор Java (javac);
- Стандартні бібліотеки класів Java;
- Приклади;
- Повну документацію;
- Різноманітні утиліти;
- Виконавчу систему Java (JRE[10]).

В склад JDK не входить інтегроване середовище розробки на Java (IDE), тому програміст, який використовує тільки JDK, повинен також використовувати текстовий редактор і компілювати свої програми через командний рядок.

2.3 IntelliJ IDEA Community Edition

Для написання програм мовою програмування Java достатньо і звичайного текстового редактору, але такий варіант не може конкурувати з використанням професійних середовищ розробки – IDE.

IntelliJ IDEA – це інтегроване середовище розробки програмного забезпечення мовою Java від компанії JetBrains, раніше відомої як IntelliJ.

Перша версія програми з'явилась в 2001 році і являлась першим середовищем розробки для Java. Дизайн середовища див.рис.2.3 орієнтований на продуктивність роботи програмістів.

IntelliJ IDEA випускається в двох варіантах: Community Edition та Ultimate Edition. Перша версія є повністю безкоштовною, друга версія є платною. В даному дипломному проєкті використовується саме IntelliJ IDEA Community Edition.

Програма містить в собі повний набір компонентів необхідних для створення додатків: редактор, середовище компіляції та виконання.

Варто відмітити, що платформа IntelliJ розроблена в JetBrains, стала основою для продукту від компанії Google – Android Studio, середовища розробки під пристрої з операційною системою Android.

2.4 Інструментарій JavaFX

Одним із головних мінусів віртуальної машини Java протягом багатьох років являлась повна відсутність нормальних бібліотек для створення інтерфейсу. Хоча SWT та Swing являють собою потужні системи для створення та відображення графічного інтерфейсу, все ж їх неможливо назвати швидкими та сучасними.

4 грудня 2008 року американська компанія Sun Microsystem випустила першу платформу для створення інтерфейсу – JavaFX.

Варто відмітити, що JavaFX – програмна платформа для створення функціонально багатих додатків для різних операційних систем: Windows, MacOS, Linux та для різних пристроїв:

- PC;
- Смартфони;
- Планшети;
- Вбудовані пристрої;

- ТБ.

Додатки з використанням JavaFX будуть працювати всюди, де встановлена виконавча система Java (JRE)

Варто відмітити, що JavaFX ще й окрема мова програмування. В компанії Sun Microsystem для даної мови використовують спеціальну назву – JavaFX Script, хоча ця мова і не являється скриптовою мовою, а відразу компілюється в байт-код Java.

Незважаючи на відмінності мов, платформа JavaFX інтегрована з середовищем JRE, а робоче середовище JavaFX інтегроване в пакет завантаження Java, тому установка робочого середовища JavaFX є частиною процесу установки JRE.

JavaFX являється інструментарієм наступного покоління для створення графічного інтерфейсу, який дозволяє розробляти багатоплатформені додатки, які коректно себе поведуть незалежно від платформ на яких вони запускаються.

JavaFX призначений для забезпечення додатків з графічним інтерфейсом такими складними функціями як плавна анімація, відтворення аудіо та відео завдяки використанню апаратного прискорення графіки та можливостей GPU.

З допомогою JavaFX можна створювати програми для різних операційних систем: Windows, MacOS, Linux та для різних пристроїв:

- PC;
- Смартфони;
- Планшети;
- Вбудовані пристрої;
- ТБ.

Додатки з використанням JavaFX[11] будуть працювати всюди, де встановлена виконавча система Java (JRE).

JavaFX надає великі можливості в порівнянні з рядом інших подібних платформ, зокрема, в порівнянні зі платформою Swing, це і великий набір елементів управління, і можливості по роботі з мультимедіа, двомірної і тривимірною графікою, декларативний спосіб опису інтерфейсу за допомогою мови розмітки FXML, можливість стилізації інтерфейсу за допомогою CSS і багато іншого.

Середовищем візуальної розробки виступає JavaFX Scene Builder. Він являється ключовим засобом розробки для платформи JavaFX. JavaFX Scene Builder дозволяє швидко створювати графічні інтерфейси для додатків JavaFX без необхідності написання коду. Коли користувач створює макет інтерфейсу, автоматично генерується відповідний код на мові FXML.

JavaFX Scene Builder представляє досить простий, інтуїтивно зрозумілий інтерфейс див. рис. 2.2, який може допомогти швидко створити прототип додатку.

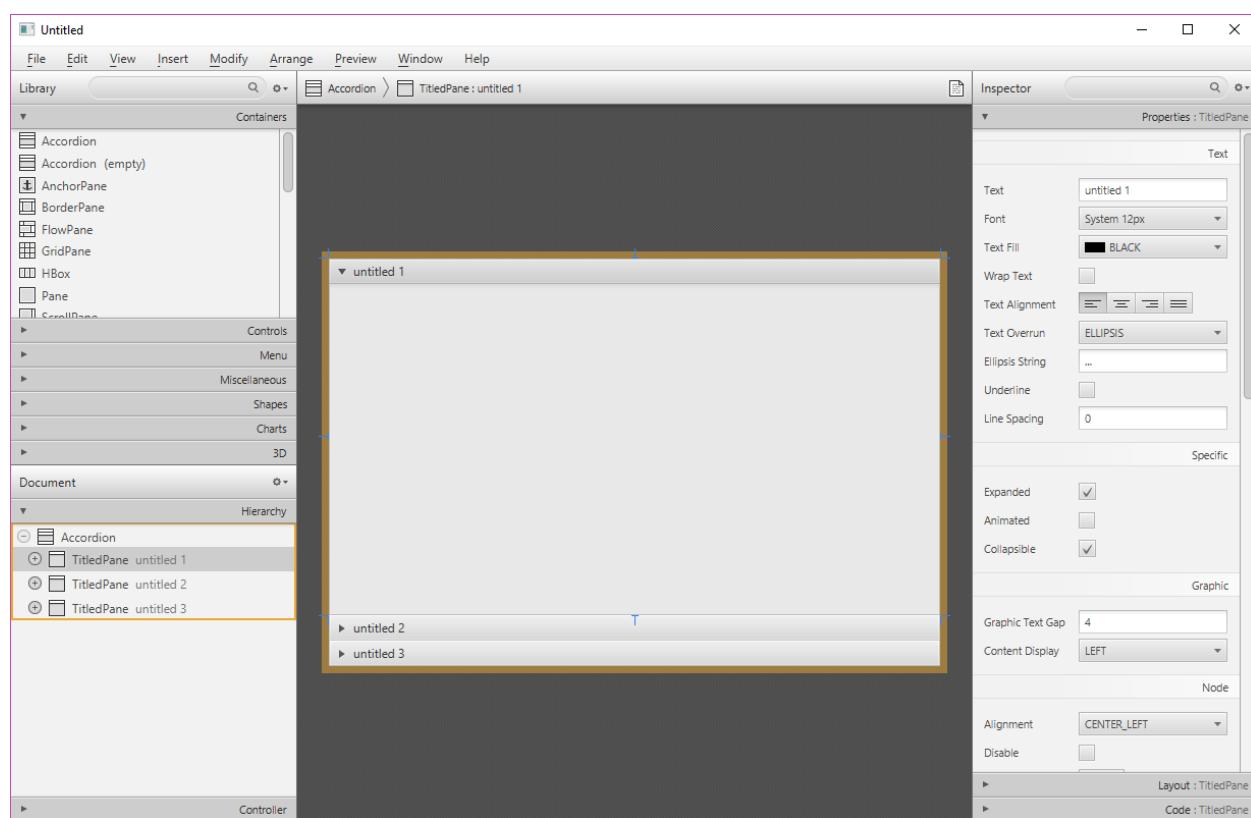


Рисунок 2.1 – інтерфейс JavaFX Scene Builder.

Основними характеристиками JavaFX Scene Builder є:

- Drag-and-drop[12] інтерфейс, дозволяє швидко створювати користувацький інтерфейс без необхідності написання коду.
- Проста інтеграція з будь-яким інтегрованим середовищем розробки Java.
- Автоматична генерація коду FXML відбувається при створенні та зміні елементів користувацького інтерфейсу.
- Редагування та огляд в реальному часі дозволяє швидко візуалізувати зміни в користувацькому інтерфейсі без необхідності компіляції.
- Підтримка CSS дозволяє керувати зовнішнім виглядом користувацького інтерфейсу додатка
- Підтримка багатоплатформеності забезпечується для операційних систем Windows, MacOS, Linux.
- Безкоштовність.

На даний момент JavaFX являється найкращим способом для створення графічних додатків за допомогою мови Java, який прийшов на зміну AWT і Swing. Також варто відзначити, що для роботи з JavaFX замість мови програмування Java теоретично можна використовувати будь-яку мову програмування, яка підтримується JVM.

2.5 База даних H2

Обрано оптимальний варіант бази даних - вбудовану базу даних H2 [13]. Оскільки задачею було створити безпечну систему для роботи на конкретному ПК, без необхідності постійної підтримки серверу – база даних H2, враховуючи її переваги, є найкращим варіантом для реалізації завдання на дипломний проєкт, враховуючи вихідні данні.

Розробка ядра бази даних H2 почалась в 2004 році та була вперше опублікована в 2005 році. Розробником бази даних являється Тома Мюллер, який ще раніше розробив механізм бази даних Java Hypersonic SQL. В 2001 році проєкт Hypersonic SQL був зупинений і група розробника почала працювати над Hypersonic 2, звідки і пішла назва бази даних – H2. Не зважаючи на схожість в назві та ту ж саму команду, база даних H2 була побудована з нуля.

H2 Database– це система керування реляційними базами даних, яка написана мовою програмування Java. Вона може виступати в якості вбудованої в бази даних або може бути запущена в режимі клієнт-сервер.

Незважаючи на багату функціональну здатність бази даних, при її розробки не використовувались сторонні бібліотеки. Це дало змогу значно зменшити розміри СУБД. Вся СУБД вміщується на комп'ютері в файлі розміром 1МБ, а весь архів, включаючи приклади та вихідний код – в 5МБ. Для порівняння СУБД Oracle 11 для Microsoft Windows займає приблизно 1.7ГБ.

Однією з цікавих властивостей у H2 це можливість створення бази даних в оперативній пам'яті (In Memory Database), замість того, щоб зберігати інформацію на жорсткому диску комп'ютера. Це робить швидкість запитів та всі можливі маніпуляції з даними дуже швидкими. Варто відмітити, що при виборі властивості «In Memory Database» дані існують тільки до тих пір, поки працює програма чи додаток, коли програма чи додаток припиняють свою роботу, всі дані, які були збережені в базі даних, будуть видалені із пам'яті.

Для створення СУБД було обрано мову програмування Java по наступним причинам:

- Багатоплатформеність Java
- Підтримка Unicode
- Проста інтеграція з Java додатками
- Користувачькі функції працюють дуже швидко

JDBC - Java DataBase Connectivity – це стандартний API для незалежного з’єднання мови програмування Java з різними базами даних. Головна роль JDBC - автоматичне створення з’єднань з базою даних. Загалом, JDBC це бібліотека, яка забезпечує купу різноманітних інтерфейсів для роботи з різними базами даних.

Для H2 основними API – інтерфейсами являється SQL та JDBC, але база даних також підтримує використання драйвера PostgreSQL ODBC, виконуючи роль сервера PostgreSQL[14].

H2 надає адміністративний інструмент під назвою H2 Console див. рис. 2.1, з яким через браузер і може працювати програміст.

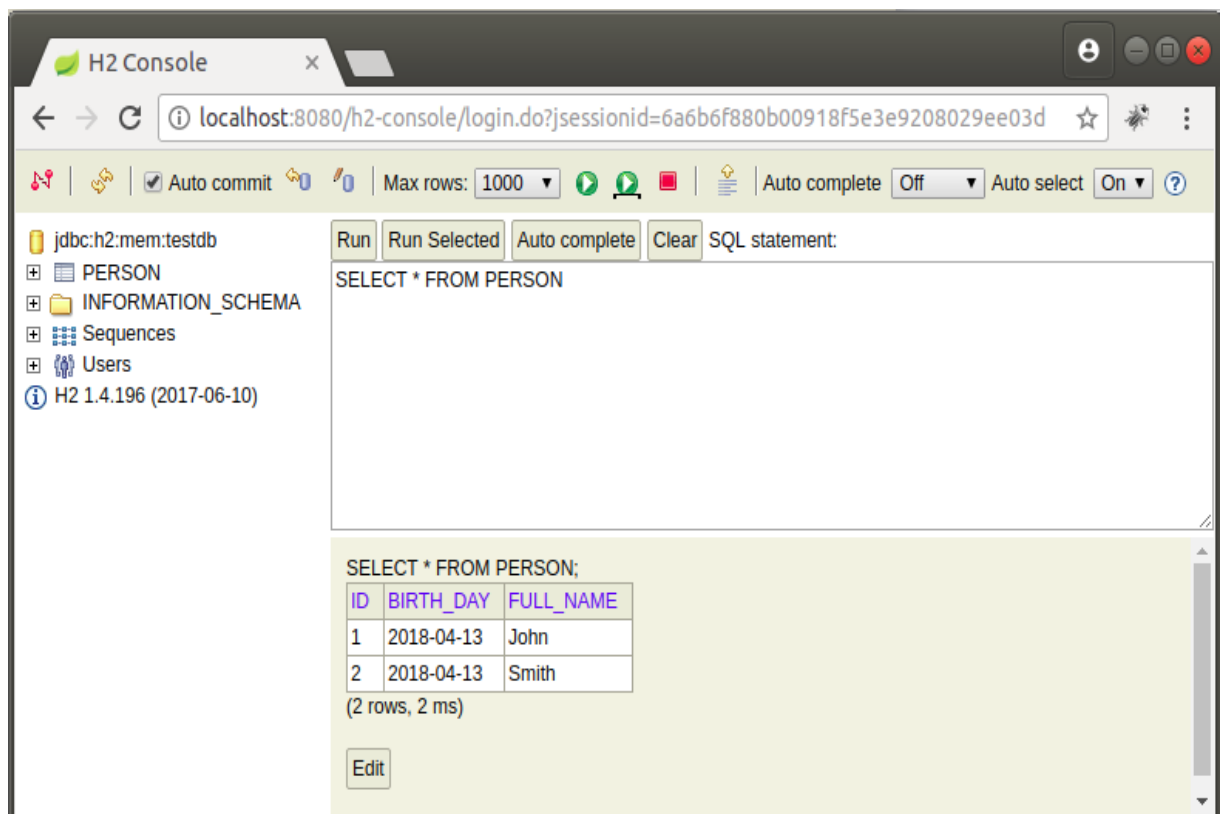


Рисунок 2.2 - H2 Console

Щодо безпеки обраної бази даних, вона підтримує захист від SQL-ін'єкцій[15] шляхом примусового використання параметризованих операторів. В H2 ця функція називається «вимкнення літералів».

У вбудованій версії бази даних SQL-ін'єкції не є основною загрозою, але при використанні бази даних в режимі клієнт-сервер, SQL-ін'єкції – одна з основних проблем. Укорінення SQL-коду або SQL-ін'єкції – один із найрозповсюджених способів злому сайтів чи додатків, які працюють з базами даних. SQL-ін'єкції ґрунтуються на укоріненні в запит довільного SQL-коду. Укорінення, в залежності від типу використовуваної СУБД, може дати можливість атакуючому виконати запит до бази даних. Наприклад можна прочитати вміст будь-яких таблиць, видалити, замінити чи додати нові дані, можна отримати можливість читання та/або запису локальних файлів, виконання довільних команд на сервері, що знаходиться під атакою.

2.6 Бібліотека Apache POI

Завершальним етапом роботи даного програмного забезпечення є створення документів в форматі .doc з введеною користувачем інформацією з можливістю подальшого друку. В мові програмування Java немає бібліотек для обробки чи роботи з документами в форматі .doc, тому для виконання даної функції, було використано додаткову бібліотеку Apache POI від компанії Apache Software Foundation

Apache POI – це бібліотека мови програмування Java з відкритим вихідним кодом. Раніше дана бібліотека була частиною проєкту Jakarta. Бібліотека допомагає працювати з такими документами компанії Microsoft як Word, Excel, Power point та інші[16].

POI – в даному випадку це аббревіатура «Poor Obfuscation Implementation» тобто «Погана заплутана реалізація». В процесі розробки програми Apache POI у інженерів Apache виникла купа запитань до реалізації форматів компанією Microsoft, саме тому назви всіх бібліотек Apache POI мають жартівливу форму.

В Apache POI є окремі компоненти для роботи з кожним форматом від компанії Microsoft див. табл. 2.3[17].

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

Таблиця 2.3 – Опис компонентів Apache POI.

Префікс компонента	Повна назва компонента	Описання компонента	Формат
POIFS	«Poor Obfuscation Implementation File System» або «Погана заплутана реалізація файлової системи»	Базовий керуючий компонент для всіх елементів POI	
HSSF	«Horrible Spreadsheet Format» або «Жахливий формат електронних таблиці»	Компонент для зчитування та запису файлів MS-Excel	XLS
XSSF	«XML Spreadsheet Format» або «XML формат електронних таблиць»	Компонент для зчитування та запису файлів MS-Excel	XLSX
HPSF	«Horrible Property Set Format» або «Жахливий формат набору властивостей»	Компонент для отримання наборів властивостей файлів MS-Office	
HWPB	«Horrible Word Processor Format» або «Жахливий формат текстового процесора»	Компонент для зчитування та запису файлів MS-Word	DOC

Продовження таблиці 2.3 – Опис компонентів Apache POI.

XWPF	«XML Word Processor Format» або «XML Формат текстового процесора»	Компонент для зчитування та запису файлів MS-Word	DOCX
HSLF	«Horrible Slide Layout Format» або «Жахливий формат макета слайдів»	Компонент для зчитування та запису файлів PowerPoint	PPT
XSLF	«XML Slide Layout Format» або «XML формат слайдів»	Компонент для зчитування та запису файлів PowerPoint	PPTX
HDGF	«Horrible DiaGram Format» або «жахливий формат діаграм»	Компонент для роботи з файлами MS-Visio	VSD
XDGF	«XML DiaGram Format» або «XML формат діаграм»	Компонент роботи з файлами MS-Visio	VSDX
HPBF	«Horrible PuBlisher Format» або «Жахливий видавницький формат»	Компонент для зчитування та запису файлів MS-Publisher	XPS
DDF	«Dreadful Drawing Format» або «Жахливий формат малювання»	Компонент для зчитування та запису файлів Microsoft Office Drawing	DOC

Для реалізації даного проєкту нам необхідні такі компоненти:

- POIFS – для роботи з усіма доступними компонентами Apache POI
- компонент HWPF – для запису інформації в файли формату .doc.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		32

Висновки до розділу

Наразі існує купа різноманітних мов програмування і кожна з них має свої плюси та мінуси. Враховуючи специфіку проєкту, та відсутність обмежень в виборі засобів програмування, для реалізації цього проєкту було обрано мову програмування Java, оскільки вона має оптимальні засоби для рішення поставленої задачі.

В якості СУБД виступає H2 Database. Враховуючи обрану мову програмування Java, оптимальним рішенням було обрати базу даних, яку пропонують самі розробники мови. Для реалізації даного проєкту потрібна була вбудована база даних і H2 є ідеальним варіантом. Перевагу надано саме вбудованій базі даних оскільки така БД підходить під вимоги даного проєкту, являється безпечним варіантом для збереження даних, не потребує постійної роботи серверу та доступу в інтернет. Програми з вбудованими БД є повністю автономними.

Для запису та зчитування інформації в файли формату .doc, використано бібліотеку Apache POI, яка значно полегшує процес роботи з файлами потрібного нам формату.

В якості середовища розробки графічного інтерфейсу даного проєкту, було обрано JavaFX, оскільки дане середовище пропонують розробники мови програмування. Варто відмітити, що JavaFX в деяких моментах використовує нові методи, які відрізняються методів, що використовують аналоги.

Підбиваючи підсумки можна сказати, що для реалізації даного дипломного проєкту було обрано оптимальні засоби, які ідеально взаємодіють між собою, а саме:

- Мова програмування Java
- Інструментарій для створення графічного інтерфейсу JavaFX
- СУБД H2 Database
- Бібліотека Apache POI

3 ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ ПАЦІЄНТІВ

3.1 Розроблена архітектура програмного забезпечення

Запропоновано розробити додаток до МІС, який відповідає таким вимогам:

- Надійність збереження даних
- Зручність в користуванні
- Безвідмовна робота
- Швидкодія програми
- Простий та зрозумілий інтерфейс
- Коректність запису даних в текстові документи

В процесі розробки програмного забезпечення, в рамках поставленої задачі, була розроблена архітектура створюваного програмного забезпечення, яка зображена на додатку 1.

Інтерфейс користувача – це інтерфейс з яким взаємодіє користувач, при роботі з даним програмним забезпеченням. В інтерфейсі відображатимуться усі поля, які заповнить користувач для занесення інформації в базу даних. Також відображатимуться результати, які користувач запросить з бази даних.

Блок пам'яті – у цей блок входить база даних. В створеній базі даних будуть зберігатимуться всі дані, які користувач вводить в програму, а також логіни та паролі для входу до системи.

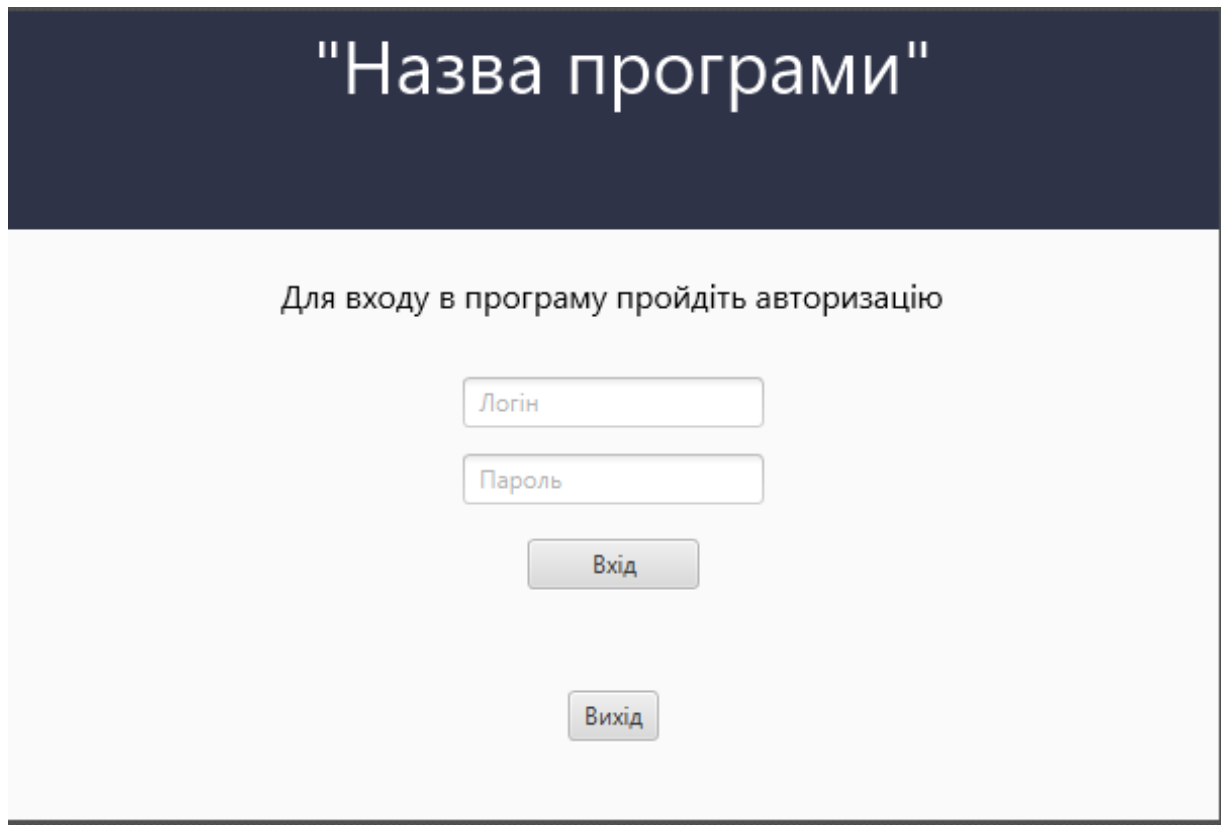
Процесинговий блок – тіло програми, в якому відбуватимуться всі необхідні процеси та всі необхідні обчислення, процес виводу результату на інтерфейс користувача а також процес створення документів, які згодом підуть на друк.

3.2 Розробка інтерфейсу користувача

Для максимально спрощеної роботи з програмою постала задача – створити інтуїтивно зрозумілий інтерфейс без зайвих опцій, в якому було б дуже легко освоїтись людині, яка ніколи не працювала зі схожими програмами.

Для реалізації інтерфейсу була проведена консультація з замовником даного проєкту, щоб зрозуміти, який загальний вигляд користувацького інтерфейсу буде комфортним для використання.

При запуску додатку перед користувачем постає вікно авторизації див. рис. 3.1, в якому користувачу пропонується ввести свій логін та пароль.



"Назва програми"

Для входу в програму пройдіть авторизацію

Логін

Пароль

Вхід

Вихід

Рисунок 3.1 – Вікно авторизації в програму.

Після чого програма перевіряє чи існує користувач з таким логіном та паролем в базі даних зареєстрованих користувачів. Блок-схема роботи вікна авторизації наведена в Додатку 2.

Після успішної авторизації користувач попадає в вікно вибору подальшого режиму роботи програми. Щоб уникнути плутанини, подальшу роботу програми поділено на дві робочі частини, які користувач обирає самостійно див.рис.3.2.

Перша частина виконується при натисканні кнопки «Додати історію пацієнта» див. рис. 3.3. В цій частині виконується робота з новими даними, створюється нова історія хвороби пацієнта, вводяться нові дані, які треба обробити та занести в базу даних для подальшого зберігання.

Друга частина виконується при натисканні кнопки «Робота з існуючою історією». В цій частині виконується робота вже з існуючими даними тобто доповнення існуючої інформації, її виведення, редагування, перевірка, пошук в базі даних тощо.

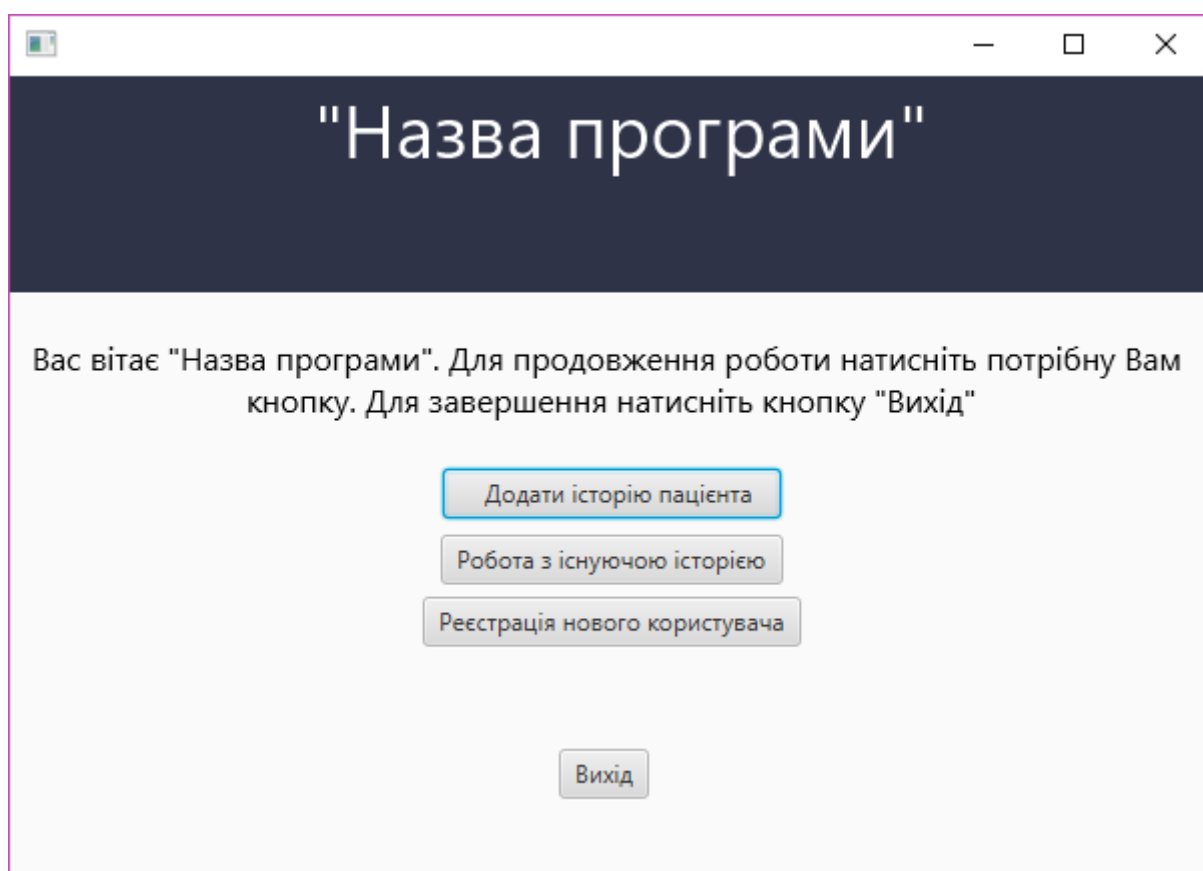


Рисунок 3.2 – Меню вибору режиму роботи програми.

Також у вікні вибору режиму роботи програми присутня кнопка «Реєстрація нового користувача», при натисканні на неї викликається вікно, в якому з'являється можливість занести логін та пароль нового користувача до бази даних зареєстрованих користувачів програми див.рис.3.3.

"Назва програми"

Для входу в програму пройдіть авторизацію

Рисунок 3.3 – Вікно реєстрації нового користувача

Натискання кнопки «Додати історію пацієнта» викликає вікно див.рис.3.4, де користувач заповнює первинну інформацію про пацієнта, загальні данні, результати первинного огляду та діагноз з яким пацієнт поступив до медичного закладу і загалом призводить до створення нової історії хвороби в базі даних.

Після введення потрібних даних в відповідні поля, користувач має натиснути кнопку «Зберегти» цим самим записавши всю введену інформацію в базу даних.

Після збереження інформації користувач може продовжити роботу в поточному вікні або виконати наступні дії:

- Натиснути кнопку «Продовжити роботу», цим самим перейти до наступного вікна введення детальної інформації.

- Натиснути кнопку «До попереднього вікна» і перейти до попереднього вікна вибору режиму роботи програми.
- Натиснути кнопку «Вийти» і повністю завершити виконання програми.

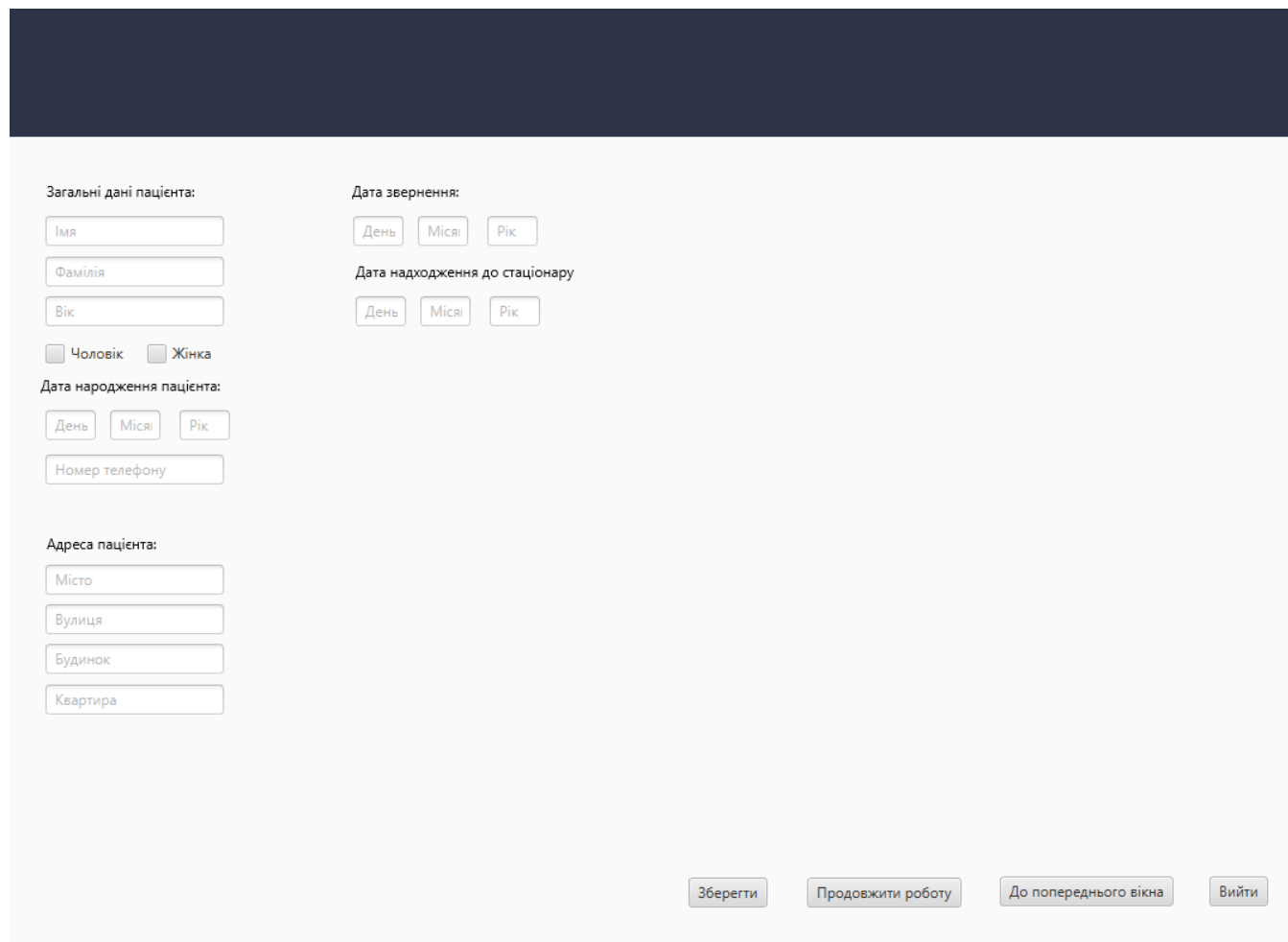


Рисунок 3.4 – Вікно введення базової інформації.

При натисканні кнопки «Продовжити роботу», вікно введення базової інформації закривається і відкривається вікно введення детальної інформації див.рис.3.5. Рішення розділити вікна було прийняте в зв'язку з великою кількістю полів, які потрібно заповнити. В той же час при розробці інтерфейсу була задача не захламляти робоче вікно великою кількістю полів, та залишити розмір самого вікна досить компактним. Варто відмітити, що детальна інформація заповнюється вже під час перебування хворого в стаціонарі по мірі його обстеження.

Рисунок 3.5 – Вікно введення детальної інформації.

Після введення потрібних даних в відповідні поля, користувач знову має натиснути кнопку «Зберегти» цим самим записавши всю нову введену інформацію в базу даних.

Для подальшої роботи користувачу пропонуються вибір:

- Натиснути кнопку «Продовжити роботу», цим самим перейти до наступного вікна вибору документів, які потрібно створити і в подальшому роздрукувати.
- Натиснути кнопку «До попереднього вікна» і перейти до попереднього вікна введення базової інформації.
- Натиснути кнопку «Вийти» і повністю завершити виконання програми.

При натисканні кнопки «Продовжити роботу», вікно введення додаткової інформації закривається і відкривається вікно вибору бланків див.рис.3.6.

Рисунок 3.6 – Вікно вибору бланків.

При своїй роботі лікарі використовують велику кількість різноманітних бланків для виписок та направлень. Кількість та їх тип залежить від конкретного пацієнта та його хвороби. В даному проєкті реалізоване вікно вибору потрібних бланків та документів, в залежності від кожного пацієнта.

При відкритті вікна, користувачу пропонується обрати один або декілька бланків в залежності від ситуації. Вибір потрібних бланків виконується шляхом натискання користувача на «прапорець» навпроти відповідного бланку.

Після вибору потрібних бланків, користувачу пропонуються вибір:

- Натиснути кнопку «Створити файли», в результаті чого, програма створить в папці файли з інформацією, яку попередньо ввів користувач.

- Натиснути кнопку «До попереднього вікна» і перейти до попереднього вікна введення додаткової інформації.
- Натиснути кнопку «Вийти» і повністю завершити виконання програми.

При натисканні кнопки «Робота з існуючою історією» перед користувачем відкривається вікно з пацієнтами, які вже є в базі даних див.рис.3.7. У користувача є змога відсортувати базу даних по існуючим критеріям вгорі таблиці або ввести в поле пошуку будь-яку інформацію, за якою користувач хоче отримати список пацієнтів.

При виборі потрібного пацієнта з списку, у користувача є змога подивитися його данні та при необхідності їх відредагувати.

Після вибору пацієнта з списку, користувачу пропонуються вибір:

- Натиснути кнопку «Переглянути інформацію», в результаті чого, буде відкрите вікно введення базової інформації див.рис.3.4 з усіма його властивостями та кнопками, але поля з інформацією будуть заповнені відповідними даними з БД.
- Натиснути кнопку «До попереднього вікна» і перейти до попереднього вікна вибору режиму роботи програми.
- Натиснути кнопку «Вийти» і повністю завершити виконання програми.

имя	Фамилия	адрес
No content in table		
+		

Удалить

Закрыть

Рисунок 3.7 – Вікно вибору історій пацієнтів.

Висновки до розділу

В даному розділі була проведений детальний розбір функціоналу додатку, показана послідовність виконання та викликів вікон програми.

В даному проекті було використано принцип послідовного виклику вікон, щоб інтерфейс був простий в користуванні, зрозумілий та підходив під вимоги користувача.

Було реалізовано функцію авторизації та реєстрації користувача в додатку, щоб забезпечити надійність існуючих даних пацієнтів.

ВИСНОВКИ

В даному дипломному проекті було розглянуто існуючі медичні системи, які впроваджуються на території України.

Незважаючи на те, що зараз масово впроваджується системи медичних інформаційних систем, сфера їх застосування охоплює тільки первинну медицину – сімейних лікарів, терапевтів, педіатрів. Решта лікарів досі використовує паперові носії під час своєї роботи.

Сучасні МІС впроваджуються на рівні закладів та зовсім не пристосовані для роботи лікарів-хірургів, тож задачею дипломного проекту було створити систему, яка компенсувала мінуси існуючих систем, була зрозумілою в освоєнні та була б розрахована на індивідуальне використання конкретного лікаря-спеціаліста.

Ключовими вимогами до створення МІС являються: надійність збереження даних; зручність в користуванні; безвідмовна робота; швидкодія програми; простий та зрозумілий інтерфейс.

Враховуючи вимоги до створення МІС а також особисті побажання лікаря-хірурга, була запропонована система, яка описується в даному дипломному проекті. Система має інтуїтивно зрозумілий та приємний інтерфейс без зайвих функцій. Для системи було використано вбудовану систему даних, оскільки для реалізації поставленої задачі така база даних оптимально підходила.

Система пройшла тестування та використовується за призначенням лікарем-спеціалістом, про що свідчить довідка про впровадження, яка наведена в додатку 5.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		44

ПЕРЕЛІК ПОСИЛАНЬ

1. <http://moz.gov.ua/article/reform-plan/teper-oficijno-moz-ukraini-rozpochinae-pershij-etap-transformacii-ekstrenoi-medichnoi-dopomogi->
2. Закон України від 01.06.2010, № 2297-VI "Про захист персональних даних"
3. Порядок функціонування електронної системи охорони здоров'я, затверджений Постановою КМ України, від 25.04.2018, № 411 "Деякі питання електронної системи охорони здоров'я"
4. <https://public.docs.openprocurement.org/get/3cdf2df19b74f10a0783132598712a7?KeyID=52462340&Signature=ZDVuu33lp01L4AFMcJNnWaknVo3Cc1%2FmbvhFAgjJDN9OtybMv6HX3ujL%252B1%2Fs5PnWgH%252B2oBYXXyng9GXJgMqwBA%253D%253D>
5. <https://helsi.me/>
6. <https://www.mcmed.ua/ua>
7. <https://www.oracle.com/technetwork/java/index.html>
8. <https://www.baeldung.com/java-stack-heap>
9. <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
10. <https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
11. <http://www.h2database.com/html/functions.html>
12. https://docs.oracle.com/javafx/2/drag_drop/jfxpub-drag_drop.htm
13. <https://docplayer.ru/27558273-Razrabotka-nasyshchennyh-internet-prilozheniy-s-pomoshchyu-javafx.html>
14. <https://www.theserverside.com/discussions/thread/46456.html>
15. https://www.w3schools.com/sql/sql_injection.asp
16. http://www.tutorialspoint.com/apache_poi/apache_poi_overview.htm
17. <https://poi.apache.org/>

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		45

ДОДАТКИ

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		46